



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/671,180	09/24/2003	Gundrala D. Goud	42P16557	7211

7590 03/17/2008  
Cory G. Claassen  
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP  
Seventh Floor  
12400 Wilshire Boulevard  
Los Angeles, CA 90025

EXAMINER
----------

CHEA, PHILIP J

ART UNIT	PAPER NUMBER
----------	--------------

2153

MAIL DATE	DELIVERY MODE
-----------	---------------

03/17/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/671,180	<b>Applicant(s)</b> GOUD ET AL.	
	<b>Examiner</b> PHILIP J. CHEA	<b>Art Unit</b> 2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 26 December 2007.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)                                | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

## DETAILED ACTION

## DETAILED ACTION

This Office Action is in response to an Amendment filed July 30, 2007. Claims 1-30 are currently pending. Any rejection not set forth below has been overcome by the current Amendment.

### ***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1,7-9,11, and 17-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Yang et al (Pub # US 2004/0024831), and further in view of IPMI (Intelligent Platform Management Bus Communications Protocol Specification) and further in view of Emerson et al. (US 2003/0131136), herein referred to as Emerson.

As per claim 1, Yang et al teach a method of activating a management mode of operation of a processor on a processing blade (see e.g. [0003], lines 1-3, which teaches limitation because the prior art embodies a method for the management of a blade server including hardware, such as a processor or a server blade), the processing blade included within a blade server (see e.g. [0003], lines 1-5, which teach this limitation because server blades are included with the blade server), interacting with a management module of the blade server during the management mode of operation to manage operation of the processing blade (see e.g. [0007], lines 1-5, which teaches this limitation because the management blade is in conjunction with the blade server during the management of each server blade), and deactivating the management mode of operation of the processor (see e.g. [0010], lines 13-15 which teaches this limitation because a slave management blade is used to replace a master management blade upon failure of the blade).

Art Unit: 2153

Although the system disclosed by Yang shows substantial features of the claimed invention (discussed above), it fails to disclose using a firmware unit and a software proxy layer to emulate a baseboard management controller.

Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Yang, as evidenced by IPMI and Emerson.

In an analogous art, IPMI discloses a server management system that allows a firmware unit to interact with a management module of a server (see page 6, "1.6 Platform Management Network Topology", describing how firmware can be used to embed the management messaging protocol and other baseboard-specific functions).

Given the teaching of IPMI, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Yang by employing a firmware, such as disclosed by IPMI, in order to provide embedded server management functions.

Although the system disclosed by Yang in view of IPMI shows substantial features of the claimed invention (discussed above), it fails to disclose that a software proxy layer to emulate a baseboard management controller.

Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Yang in view of IPMI, as evidenced by Emerson.

In an analogous art, Emerson discloses a remote server management controller that is adapted to be incorporated into a managed server so that a user may obtain management data from the OS or perform other functions remotely by causing the managed server to redirect data from its local communications interfaces to communications interfaces that are available to the remote user (see Abstract). Emerson further discloses a software proxy layer to emulate a baseboard management controller (see paragraph 68, *describing a VCD (i.e. software proxy layer) which is a virtual communication device that allows the remote server management controller to communication with specific OS features such as the Emergency Management Services* and paragraphs 73-74, *describing how the VCD emulates a baseboard management controller in the form of intercepting signals from the*

Art Unit: 2153

*system management controller and emulating serial ports that the OS locates for the use of the Emergency Management Services).*

Given the teaching of Emerson, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Yang in view of IPMI by employing a software proxy layer to emulate a baseboard management controller, such as disclosed by Emerson, in order to monitor and correct failure conditions in a networked computer system.

With respect to claim 7, Yang et al teach a method of activation of activating a plurality of management modes of operation of a corresponding plurality of processing blades of the blade server in response to a corresponding plurality of software entities executing on each of the processing blades (see e.g. [0017], lines 27-31, which teaches this limitation because the implementations of the master management blade of the blade server for managing switch and server blades occurs via applications software requests for software being controlled by each of the server and switch blades), interacting with the management module of the blade server during each of the plurality of management modes of operation to manage operation of each of the plurality of processing blades (see e.g. [0008], lines 1-6, which teaches this limitation because a middle interface within the blade management system to manage each blade within the blade server system), and deactivating the plurality of management modes of operation (see e.g. [0010], lines 13-15, which teaches this limitation because a slave management blade is used to replace a master management blade upon failure of the blade).

With respect to claim 8, Yang et al teach a method of activating the plurality of management modes of operation, which comprises activating each one of the plurality of management modes of operation of each of the plurality of processing blades at an independent time (see e.g. [0008], lines 1-10, which teaches this limitation because the master management blade is implemented to control each of the server and switch blades within the blade server management system directly and independent from one another, as shown in sec. [0003], lines 14-16) and deactivating the plurality of management modes of operation, which comprises deactivating each one of the plurality of management modes of each of the plurality of processing blades at an independent time (see e.g. [0010], lines 13-15, which teaches this

Art Unit: 2153

limitation because the master management blade, used to control the various switch and server blades, loses control of said server and switch blades upon failure and the slave management blade becomes the new master management blade as each server blade works independently from other blades).

With respect to claim 9, Yang et al teach a method of interacting with the management module includes at least one of coordinating with the management module for access to shared resources of the blade server, reporting system errors to the management module (see e.g. [0005], lines 10-14, which teaches this limitation because management server is implemented to report any unusual functioning of any of the server blades within the blade server) and coordinating fault resilient booting with the management module (see e.g. [0006], lines 13-15, which teaches this limitation because server managers repair crashed blade servers to revamp the stability of the blade server).

With respect to claim 11, Yang et al in view of IPMI teach a method of activating a management mode of operation of a processing blade of a blade server (see e.g. [0003], lines 1-3, which teaches limitation because the prior art embodies a method for the management of a blade server including hardware, such as a processor or a server blade), using a firmware unit (see discussion above regarding claim 1) to interact with a chassis management module ("CMM") of the blade server during the management mode of operation to manage operation of the processing blade (see e.g. [0017], lines 15-19, which teaches this limitation because the management blade includes chassis identification code information to manage blades hosted by the chassis and is in conjunction with the blade server during the management of each server blade), and deactivating the management mode of operation (see e.g. [0010], lines 13-15 which teaches this limitation because a slave management blade is used to replace a master management blade upon failure of the blade).

With respect to claim 17, Yang et al in view of IPMI teach a method of activating the plurality of management modes of operation, which comprises activating each one of the plurality of management modes of operation of a plurality of processing blades of the blade server in response to a corresponding plurality of software entities stored in or on a corresponding plurality of firmware units (see discussion above regarding claim 1) executing on each of the processing blades (see e.g. [0017], lines 27-31, which teaches this limitation because the implementations of the master management blade of the blade server

Art Unit: 2153

for managing switch and server blades occurs via applications software requests for software being controlled by each of the server and switch blades), interacting with the management module of the blade server during each of the plurality of management modes of operation to manage operation of each of the plurality of processing blades (see e.g. [0008], lines 1-6, which teaches this limitation because a middle interface within the blade management system to manage each blade within the blade server system, note that the management blade acts as the chassis management module by storing chassis identification code information to manage blades hosted by the chassis, as shown in sec. [0017], lines 15-19), and deactivating the plurality of management modes of operation (see e.g. [0010], lines 13-15, which teaches this limitation because a slave management blade is used to replace a master management blade upon failure of the blade).

With respect to claim 18, Yang et al teach a method of interacting with the CMM, which includes at least one of coordinating with the CMM for access to shared resources of the blade server (see e.g. [0003], lines 1-8 which teaches this limitation because the server blade is a resource of the blade server and are accessed by the management blades of the server manager, note that the management blade of the server manager acts as the chassis management module by storing chassis identification code information to manage blades hosted by the chassis, as shown in sec. [0017], lines 1-5), reporting system errors to the CMM (see e.g. [0005], lines 10-14, which teaches this limitation because management server is implemented to report any unusual functioning of any of the server blades within the blade server, note that the management blade of the server manager acts as the chassis management module by storing chassis identification code information to manage blades hosted by the chassis, as shown in sec. [0017], lines 15-19), and coordinating fault resilient booting with the CMM (see e.g. [0006], lines 13-15, which teaches this limitation because server managers repair crashed blade servers to revamp the stability of the blade server, note that the management blade of the server manager acts as the chassis management module by storing chassis identification code information to manage blades hosted by the chassis, as shown in sec. [0017], lines 15-19).

Art Unit: 2153

3. Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Huang et al (Pat # 6,931,475), further in view of IPMI (Intelligent Platform Management Bus Communications Protocol Specification) and further in view of Emerson et al. (US 2003/0131136), herein referred to as Emerson.

Huang et al teach a processing blade comprising a processor to execute instructions (see spec, sec. 1, lines 40-42, which teaches this limitation because a processor is embedded into each blade server), a communication link communicatively coupled to the processor (see spec, sec. 2, lines 18-27, which teaches this limitation because an output port is connected to the blade server to allow a processor to transmit information to a management board), the communication link to communicatively couple to a chassis management module of a blade server (see spec, sec. 3, lines 14-16, which teach this limitation because the information is sent from a processor to the management board, which manages the blades inserted in the sockets of each chassis), a unit communicatively coupled to the processor and having stored therein a virtual management controller (see spec, sec. 3, lines 26-32, and 62-67, which teaches this limitation because a KVM switch that is connected to the select button and the processor of the blade server, has imbedded implementations to control each blade server. The prior art shows that the KVM switch is integrated with each blade server to monitor and control the system and that each switch is coupled with the circuits on a chassis to provide for the transmission of signals to the management board, which manages the sockets of each chassis, as shown in sec. 2, lines 58-60 and sec. 3, lines 2-11), and a processor to execute the VMC to communicate with the CMM during a management mode of operation of the processor to coordinate operation of the processing blade with the CMM (see spec, sec. 2, lines 58-60 and sec. 3, lines 2-11, which teach this limitation because a KVM switch is integrated with each blade server to monitor and control the system and that each switch is coupled with the circuits on a chassis to provide for the transmission of signals to the management board, which manages the sockets of each chassis).

Although the system disclosed by Huang shows substantial features of the claimed invention (discussed above), it fails to disclose using a firmware unit and emulating a baseboard management controller.



Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Huang, as evidenced by IPMI and Emerson.

In an analogous art, IPMI discloses a server management system that allows a firmware unit to interact with a management module of a server (see page 6, "1.6 Platform Management Network Topology", describing how firmware can be used to embed the management messaging protocol and other baseboard-specific functions).

Given the teaching of IPMI, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Huang by employing a firmware, such as disclosed by IPMI, in order to provide embedded server management functions.

Although the system disclosed by Huang in view of IPMI shows substantial features of the claimed invention (discussed above), it fails to disclose emulating a baseboard management controller.

Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Huang in view of IPMI, as evidenced by Emerson.

In an analogous art, Emerson discloses a remote server management controller that is adapted to be incorporated into a managed server so that a user may obtain management data from the OS or perform other functions remotely by causing the managed server to redirect data from its local communications interfaces to communications interfaces that are available to the remote user (see Abstract). Emerson further discloses a software proxy layer to emulate a baseboard management controller (see paragraph 68, *describing a VCD (i.e. software proxy layer) which is a virtual communication device that allows the remote server management controller to communication with specific OS features such as the Emergency Management Services* and paragraphs 73-74, *describing how the VCD emulates a baseboard management controller in the form of intercepting signals from the system management controller and emulating serial ports that the OS locates for the use of the Emergency Management Services*).

Given the teaching of Emerson, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Huang in view of IPMI by employing a software

Art Unit: 2153

proxy layer to emulate a baseboard management controller, such as disclosed by Emerson, in order to monitor and correct failure conditions in a networked computer system.

4. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Dake et al (Pub # US 2004/0268157), further in view of IPMI (Intelligent Platform Management Bus Communications Protocol Specification) and further in view of Emerson et al. (US 2003/0131136), herein referred to as Emerson.

Dake et al teach a chassis having a chassis management module (see e.g. [0021], lines 14-17, which teach this limitation because a management module is implemented within the invention for monitoring various components of a chassis and determines if the components are in the correct chassis), a plurality of processing blades supported within the chassis (see e.g. [0017], lines 5-6 and 11-13, which teaches this limitation because each chassis contains a plurality of slots or racks which are used to store server blade module that consist of one or more server blades), each of the plurality of processing blades communicatively coupled to the CMM (see e.g. [0029], lines 5-10, which teaches this limitation because the chassis management module is coupled to a plurality of blades through a communication port), a processor to execute instructions (see e.g. [0014], lines 8-9, which teaches this limitation because each server blade includes a set of main processors), a communication link communicatively coupled to the processor (see e.g. [0016], lines 1-3, which teaches this limitation because each local service processor is connected to a communication port), the communication link communicatively coupled to the CMM (see e.g. [0029], lines 5-10, which teach this limitation because the communication ports are in use by the management module), a memory unit communicatively coupled to the processor and having stored therein a virtual management controller (see e.g. [0022], lines 5-7, which teaches this limitation because a flash memory device can be used as non volatile storage for the processor connected to each blade and the flash device stores management module (which is used to control resources of each server blade, as shown in sec. [0019], lines 9-10) table information), and the processor executing the VMC to communicate with the CMM during a management mode of operation of the processor to coordinate operation of each of the plurality of processing blades with the CMM (see e.g.

Art Unit: 2153

[0019], lines 8-11, which teaches this limitation because the management module communicates with its management module service processor, which is used to control resources shared by each server blade).

Although the system disclosed by Dake shows substantial features of the claimed invention (discussed above), it fails to disclose using a flash memory unit and emulating a baseboard management controller.

Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Dake, as evidenced by IPMI and Emerson.

In an analogous art, IPMI discloses a server management system that allows a firmware unit (i.e. flash memory unit) to interact with a management module of a server (see page 6, "1.6 Platform Management Network Topology", describing how firmware can be used to embed the management messaging protocol and other baseboard-specific functions).

Given the teaching of IPMI, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Dake by employing a firmware, such as disclosed by IPMI, in order to provide embedded server management functions.

Although the system disclosed by Dake in view of IPMI shows substantial features of the claimed invention (discussed above), it fails to disclose emulating a baseboard management controller.

Nonetheless, these features are well known in the art and would have been an obvious modification of the system disclosed by Dake in view of IPMI, as evidenced by Emerson.

In an analogous art, Emerson discloses a remote server management controller that is adapted to be incorporated into a managed server so that a user may obtain management data from the OS or perform other functions remotely by causing the managed server to redirect data from its local communications interfaces to communications interfaces that are available to the remote user (see Abstract). Emerson further discloses a software proxy layer to emulate a baseboard management controller (see paragraph 68, *describing a VCD (i.e. software proxy layer) which is a virtual communication device that allows the remote server management controller to communication with specific OS features such as the Emergency Management Services* and paragraphs 73-74, *describing how the VCD emulates a baseboard management controller in the form of intercepting signals from the*

Art Unit: 2153

*system management controller and emulating serial ports that the OS locates for the use of the Emergency Management Services).*

Given the teaching of Emerson, a person having ordinary skill in the art would have readily recognized the desirability and advantages of modifying Dake in view of IPMI by employing a software proxy layer to emulate a baseboard management controller, such as disclosed by Emerson, in order to monitor and correct failure conditions in a networked computer system.

5. Claims 2, 4 and 10 are rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson in view of Barr et al (Pub # US 2004/0030944).

In reference to claims 2, 4, and 10 Yang et al teach a limitation of activating a management mode of operation of a processor on a processing blade (see e.g. [0003], as stated above), and activation of the management mode of operation of the processor, which comprises activating the management mode of operation in response to a software entity executed by the processing blade (see e.g. [0010], lines 1-10, which teaches this limitation because a slave management blade is initialized as a master management blade upon an application software request. An application was one of the positive elements listed for a software entity in the applicant's spec).

Yang et al in view of IPMI in view of Emerson teaches all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor and a limitation for providing shared resources including at least a serial port and a parallel port.

The general concept of providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor and shared resources including at least a serial port and a parallel port are well known in the art as illustrated by Barr et al, which teach a method of a management mode of operation of the processor is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor (see e.g. [0028], lines 23-25, which teaches this limitation because control of frequency synthesizers is implemented using a transparency technique and the frequency of the blades are set upon reboot of each processor, as shown in sec. [0027], lines 8-10. The prior art reads on the claim limitation because if the management of the operating frequency of each blade is executed upon a reboot, then implementing a function for transparency for a processor blade must be executed upon a reboot and prior to the machine's next boot or start up) and shared resources including at least a serial port and a parallel port

Art Unit: 2153

(see e.g. [0024], lines 6-10, which teach this limitation because the frequency synthesizers used to manage the operating frequency of blades, control the serial and parallel pins of the processor for each server blade).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al in view of IPMI in view of Emerson to include the use of a limitation for providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and an operating system runtime of the processor and a limitation for providing shared resources including at least a serial port and a parallel port as illustrated by Barr et al in order to improve upon management blade implementations, as implied in sec. [0021], lines 8-14 of Barr et al.

6. Claim 3 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944).

In reference to claim 3, Barr et al teach a processor that is transparent to a pre-boot runtime of the processor (see e.g. [0028], as stated above).

Yang et al in view of IPMI in view of Emerson and Barr et al teach all the limitations as disclosed above except for the management mode of operation of the processor being further transparent to an operating system load sequence.

The general concept of a limitation for the management mode of operation of the processor being further transparent to an operating system load sequence is rejected under obvious design optimization because one of ordinary skill in the art would find it obvious to implement the management mode of operation to be transparent to an OS load sequence when the operation is transparent to a pre-boot runtime because the OS load sequence occurs immediately after the pre-boot operations. One of ordinary skill in the art would find it obvious to implement a limitation for a management mode of operation of the processor being further transparent to an operating system load sequence because Barr et al specifies in sec. [0028], lines 23-25, which teaches this limitation because control of frequency synthesizers is implemented using a transparency technique and the frequency of the blades are set upon reboot of each processor, as shown in sec. [0027], lines 8-10.

It would have been obvious for one of ordinary skill in the art to modify Yang et al in view of IPMI in view of Emerson to include the use of a limitation of a management mode of operation of the processor being further transparent to an operating system load sequence rather than a pre-boot runtime in order to provide for a more effective implementation of a blade server system wherein the management aspects of the system are transparent to a user, as implied in sec. [0028], lines 22-27.

7. Claim 5 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Abbondanzio et al (Pub # 2003/0188222).

In reference to claim 5 Yang et al teach a limitation of activating a management mode of operation of a processor on a processing blade (see e.g. [0003], as stated above).

Yang et al and IPMI and Emerson and Barr et al teach all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor.

The general concept of saving state information of the processor and saving an execution location of the processor prior to entering the management mode of operation is well known in the art as illustrated by Abbondanzio et al, which teach the limitation for saving state information of the processor and saving an execution location of the processor prior to entering the management mode of operation (see e.g. [0033], which implies this limitation because information regarding the status of each service processor is stored on a computer readable medium and portions of the software used for execution are stored in a volatile storage element, such as memory, as shown in e.g. [0027]), exiting the management mode of operation (see e.g. [0040], which implies this limitation because a service processor is non-responsive and unable to perform the duties implemented within the management module upon the expiration of a watchdog timer and prior to a hardware reset), loading the saved state information into the processor (see e.g. [0038], which implies this limitation because status signals of each service processor are provided to the fail over logic control associated with each processor), and returning the processor to

Art Unit: 2153

the saved execution location (see e.g. [0042], which teaches this limitation because a service processor is returned to its current state after a watchdog timer reset and a reset signal is sent to the control logic).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al and IPMI and Emerson and Barr et al to include the use of a limitation for saving state information of the processor and saving an execution location of the processor prior to entering the management mode of operation as illustrated by Abbondanzio et al in order to effectively implement a blade server system, as implied in sec. [0008], lines 12-14 of Abbondanzio et al.

8. Claim 6 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI and Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Abbondanzio et al (Pub # 2003/0188222).

In reference to claim 6 Yang et al teach a limitation of activating a management mode of operation of a processor on a processing blade (see e.g. [0003], as stated above).

Yang et al and Barr et al teach all the limitations as disclosed above except for a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM").

The general concept of a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") is well known in the art as illustrated by Abbondanzio et al, which teaches a limitation of a management mode of operation comprising one of a system management mode and a platform management mode (see e.g. [0021], lines 1-10, which teaches this limitation because a system management module is embedded within the blade management system).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al in view of IPMI in view of Emerson and Barr et al to include the use of a limitation a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") as illustrated by Abbondanzio et al in order to successfully manage server blades in a system, as implied in sec. [0007], lines 4-12 of Abbondanzio et al.

9. Claim 12 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson in view of Barr et al (Pub # US 2004/0030944).

In reference to claim 12 Yang et al teach a limitation of activating a management mode of operation of a processing blade of a blade server (see e.g. [0003], as stated above).

Yang et al teaches all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor.

The general concept of providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor and shared resources including at least a serial port and a parallel port are well known in the art as illustrated by Barr et al, which teach a method of a management mode of operation of the processor is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor (see e.g. [0028], which teaches this limitation because control of frequency synthesizers is implemented using transparency and the frequency of the blades are set upon reboot of each processor, as shown in e.g. [0027]. The prior art reads on the claim limitation because if the management of the operating frequency of each blade is executed upon a reboot, then implementing a function for transparency for a processor blade must be executed upon a reboot and prior to the machine's next boot or start up).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al in view of IPMI in view of Emerson to include the use of a limitation for providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor as illustrated by Barr et al in order to improve upon management blade implementations, as implied in sec. [0021], lines 8-14 of Barr et al.

10. Claims 13 and 14 are rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944).



In reference to claims 13 and 14, Barr et al teach a processor that is transparent to a pre-boot runtime of the processor (see e.g. [0028], as stated above) and activation of the management mode of operation of the processor, which comprises activating the management mode of operation in response to a software entity executed by the processing blade (see e.g. [0010], lines 1-10, which teaches this limitation because a slave management blade is initialized as a master management blade upon an application software request. An application was one of the positive elements listed for a software entity in the applicant's spec).

Yang et al in view of IPMI in view of Emerson and Barr et al teach all the limitations as disclosed above except for instructions that, if executed by the machine, will cause the machine to perform the operations wherein the management mode of operation of the processing blade being further transparent to an operating system load sequence.

The general concept of a limitation for instructions that, if executed by the machine, will cause the machine to perform the operations wherein the management mode of operation of the processing blade being further transparent to an operating system load sequence is rejected under obvious design optimization because one of ordinary skill in the art would find it obvious to implement the management mode of operation to be transparent to an OS load sequence when the operation is transparent to a pre-boot runtime because the OS load sequence occurs immediately after the pre-boot operations. One of ordinary skill in the art would find it obvious to implement a limitation for a management mode of operation of the processor being further transparent to an operating system load sequence because Barr et al specifies in sec. [0028], lines 23-25, which teaches this limitation because control of frequency synthesizers is implemented using a transparency technique and the frequency of the blades are set upon reboot of each processor, as shown in sec. [0027], lines 8-10.

It would have been obvious for one of ordinary skill in the art to modify Yang et al in view of IPMI in view of Emerson to include the use of a limitation of a management mode of operation of the processor being further transparent to an operating system load sequence rather than a pre-boot runtime in order to provide for a more effective implementation of a blade server system wherein the management aspects of the system are transparent to a user, as implied in sec. [0028], lines 22-27.

11. Claim 15 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Abbondanzio et al (Pub # 2003/0188222).

In reference to claim 15 Yang et al teach a limitation of activating a management mode of operation of a processing blade of a blade server (see e.g. [0003], as stated above).

Yang et al and Barr et al teach all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor.

The general concept of saving state information of the processor and saving an execution location of the processor prior to entering the management mode of operation is well known in the art as illustrated by Abbondanzio et al, which teach the limitation for saving state information of the processor and saving an execution location of the processor prior to entering the management mode of operation (see e.g. [0033], which implies this limitation because information regarding the status of each service processor is stored on a computer readable medium and portions of the software used for execution are stored in a volatile storage element, such as memory, as shown in e.g. [0027]), exiting the management mode of operation (see e.g. [0040], which implies this limitation because a service processor is non-responsive and unable to perform the duties implemented within the management module upon the expiration of a watchdog timer and prior to a hardware reset), loading the saved state information into the processor (see e.g. [0038], which implies this limitation because status signals of each service processor are provided to the fail over logic control associated with each processor), and returning the processor to the saved execution location (see e.g. [0042], which teaches this limitation because a service processor is returned to its current state after a watchdog timer reset and a reset signal is sent to the control logic).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al in view of IPMI in view of Emerson and Barr et al to include the use of a limitation for saving state information of the processor and saving an execution location of the processor prior to entering the

Art Unit: 2153

management mode of operation as illustrated by Abbondanzio et al in order to effectively implement a blade server system, as implied in sec. [0008], lines 12-14 of Abbondanzio et al.

12. Claim 16 is rejected under 35 USC 103 as being unpatentable over Yang et al (Pub # US 2004/0024831) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Abbondanzio et al (Pub # 2003/0188222).

In reference to claim 16 Yang et al teach a limitation of activating a management mode of operation of a processor on a processing blade (see e.g. [0003], as stated above).

Yang et al and Barr et al teach all the limitations as disclosed above except for providing instructions that, if executed by the machine, will cause the machine to perform the operations wherein the management mode of operation comprises one of a system management mode ("SMM") and a platform management mode ("PMM").

The general concept of a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") is well known in the art as illustrated by Abbondanzio et al, which teaches a limitation of a management mode of operation comprising one of a system management mode and a platform management mode (see e.g. [0021], lines 1-10, which teaches this limitation because a system management module is embedded within the blade management system).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Yang et al in view of IPMI in view of Emerson and Barr et al to include the use of a limitation a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") as illustrated by Abbondanzio et al in order to successfully manage server blades in a system, as implied in sec. [0007], lines 4-12 of Abbondanzio et al.

13. Claims 20-22 are rejected under 35 USC 103 as being unpatentable over Huang et al (Pat # 6,931,475) in view of IPMI in view of Emerson in view of Barr et al (Pub # US 2004/0030944).

In reference to claims 20-22 Huang et al in view of IPMI teach a limitation for providing a firmware unit (see discussion above regarding claim 19) communicatively coupled to the processor and having stored therein a virtual management controller (see spec, sec. 3, as stated above), a VMC performing at least one of coordinating with the CMM for access to shared resources of the blade server (see spec, sec. 1, lines 30-35, which teaches this limitation because the KVM switch has access too the keyboard, monitor, and mouse of the blade server and the switch of each blade is coupled to a management board through the circuit on each chassis, as shown in sec. 3, lines 8-11), and shared resources including a monitor, keyboard, and a mouse (see spec, sec. 1, lines 30-35, which teaches this limitation because the KVM switch has access too the keyboard, monitor, and mouse of the blade server).

Huang et al teaches all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor.

The general concept of providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor and shared resources including at least a serial port and a parallel port are well known in the art as illustrated by Barr et al, which teach a method of a management mode of operation of the processor is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor (see e.g. [0028], which teaches this limitation because control of frequency synthesizers is implemented using transparency and the frequency of the blades are set upon reboot of each processor, as shown in e.g. [0027]. The prior art reads on the claim limitation because if the management of the operating frequency of each blade is executed upon a reboot, then implementing a function for transparency for a processor blade must be executed upon a reboot and prior to the machine's next boot or start up).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Huang et al in view of IPMI in view of Emerson to include the use of a limitation for providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor as illustrated by Barr et al in order to improve upon management blade implementations, as implied in sec. [0021], lines 8-14 of Barr et al.

14. Claim 23 is rejected under 35 USC 103 as being unpatentable over Huang et al (Pat # 6,931,475) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Yang et al (Pub # US 2004/0024831).

In reference to claim 23 Huang et al in view of IPMI teach a limitation for providing a firmware unit (see discussion above regarding claim 19) communicatively coupled to the processor and having stored therein a virtual management controller (see spec, sec. 3, as stated above).

Huang et al and Barr et al teach all the limitations as disclosed above except for a software entity executing on a processing blade and the software entity for requesting access to at least one of the shared resources of the blade server.

The general concept of a software entity executing on a processing blade and the software entity for requesting access to at least one of the shared resources of the blade server is well known in the art as illustrated by Yang et al, which teach the limitation for a system manager and a platform management mode within a component management system (see e.g. [0019], lines 12-14, which implies this limitation because a system manager is embedded within the system that provides for platform management for the component management entity (note that blades, e.g. switching banks may be regarded as components, as shown in sec. [0022], lines 6-7 and lines 15-16).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Huang et al in view of IPMI in view of Emerson and Barr et al to include the use of a software entity executing on a processing blade and the software entity for requesting access to at least one of the shared resources of the blade server as illustrated by Yang et al in order to effectively implement a management system, as implied in sec. [0008], lines 1-5 of Yang et al.

15. Claim 24 is rejected under 35 USC 103 as being unpatentable over Huang et al (Pat # 6,931,475) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Abbondanzio et al (Pub # US 2003/0105904).

Art Unit: 2153

In reference to claim 24 Huang et al teach a limitation for providing a processor to execute the VMC to communicate with the CMM during a management mode of operation of the processor to coordinate operation of the processing blade with the CMM (see spec, sec. 2, as stated above).

Huang et al and Barr et al teach all the limitations as disclosed above except for a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM").

The general concept of a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") is well known in the art as illustrated by Abbondanzio et al, which teaches a limitation of a management mode of operation comprising one of a system management mode and a platform management mode (see e.g. [0021], lines 1-10, which teaches this limitation because a system management module is embedded within the blade management system).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Huang et al in view of IPMI in view of Emerson and Barr et al to include the use of a limitation a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") as illustrated by Abbondanzio et al in order to successfully manage server blades in a system, as implied in sec. [0007], lines 4-12 of Abbondanzio et al.

16. Claims 26-27 are rejected under 35 USC 103 as being unpatentable over Dake et al (Pub # US 2004/0268157) in view of IPMI and in view of Emerson in view of Barr et al (Pub # US 2004/0030944).

In reference to claims 26-27 Dake et al teach a limitation a processor to execute instructions (see e.g. [0014], as stated above), providing a server with a VMC controller of each of the plurality of blades to perform at least one of coordinating with the CMM for access to shared resources of the blade server, reporting system errors to the CMM, and coordinating fault resilient booting with the CMM (see e.g. [0019], lines 8-11, which teaches this limitation because management module is in coordination with its service processor to monitor and control resources of each server blade), .

Art Unit: 2153

Dake et al in view of IPMI in view of Emerson teaches all the limitations as disclosed above except for a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor.

The general concept of providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor is well known in the art as illustrated by Barr et al, which teach a method of a management mode of operation of the processor is transparent to a pre-boot runtime of the processor and to an operating system runtime of the processor (see e.g. [0028], lines 23-25, which teaches this limitation because control of frequency synthesizers is implemented using transparency and the frequency of the blades are set upon reboot of each processor, as shown in e.g. [0027], lines 8-10. The prior art reads on the claim limitation because if the management of the operating frequency of each blade is executed upon a reboot, then implementing a function for transparency for a processor blade must be executed upon a reboot and prior to the machine's next boot or start up).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Dake et al in view of IPMI in view of Emerson to include the use of a limitation for providing a management mode of operation of the processor that is transparent to a pre-boot runtime of the processor as illustrated by Barr et al in order to improve upon management blade implementations, as implied in sec. [0021], lines 8-14 of Barr et al.

17. Claims 28-29 are rejected under 35 USC 103 as being unpatentable over Dake et al (Pub # US 2004/0268157) in view of IPMI in view of Emerson and Barr et al (Pub # US 2004/0030944) and further in view of Peacock (Pub # US 2002/0016868).

In reference to claims 28-29 Dake et al teach a limitation a processor to execute instructions (see e.g. [0014], as stated above).

Dake et al in view of IPMI in view of Emerson and Barr et al teach all the limitations as disclosed above except for executing a VMC in response to a software entity, the software entity triggering

Art Unit: 2153

activation of the management mode of operation to request access to at least one of the shared resources, and the shared resources including at least one of a keyboard, monitor, and mouse.

The general concept of executing a VMC in response to a software entity and the software entity triggering activation of the management mode of operation to request access to at least one of the shared resources is well known in the art as illustrated by Peacock, which teach a limitation of executing a VMC in response to a software entity and the software entity triggering activation of the management mode of operation to request access to at least one of the shared resources (see e.g. [0002], lines 2-6, which implies this limitation because an application program of an operating system, which has control of hardware resources within the operating system, is used to maintain the shared resources within the OS. Note that the prior art reads on the claimed limitation because sec. [0025] of Goud et al, states that “software entity may be a pre-boot device driver, an extensible firmware interface application, an OS driver, an OS application, or the like”), the software entity to trigger activation of the management mode of operation to request access to at least one of the shared resources (see e.g. [0018], lines 5-7, which implies this limitation because an infrared sniffer is embedded to determine whether or not an application program has requested access to one of the shared resources), and shared resources including a monitor, keyboard, mouse, or a serial port (see e.g. [0018], lines 5-7, which teaches this limitation because one of the requested hardware resources may be a serial port).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Dake et al in view of IPMI in view of Emerson and Barr et al to include the use of a limitation for executing a VMC in response to a software entity and the software entity triggering activation of the management mode of operation to request access to at least one of the shared resources as illustrated by Peacock in order to effectively monitor hardware resources, as implied in sec. [0004], lines 1-4 of Peacock.

18. Claim 30 is rejected under 35 USC 103 as being unpatentable over Dake et al (Pub # US 2004/0268157) in view of IPMI in view of Emerson and in view of Abbondanzio et al (Pub # US 2003/0105904).



In reference to claim 30 Dake et al teach a limitation a processor to execute instructions (see e.g. [0014], as stated above).

Dake et al teaches all the limitations as disclosed above except for a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM").

The general concept of a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") is well known in the art as illustrated by Abbondanzio et al, which teaches a limitation of a management mode of operation comprising one of a system management mode and a platform management mode (see e.g. [0021], lines 1-10, which teaches this limitation because a system management module is embedded within the blade management system).

It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Dake et al in view of IPMI in view of Emerson to include the use of a limitation a management mode of operation comprising one of a system management mode ("SMM") and a platform management mode ("PMM") as illustrated by Abbondanzio et al in order to successfully manage server blades in a system, as implied in sec. [0007], lines 4-12 of Abbondanzio et al.

### ***Response to Arguments***

19. Applicant's arguments with respect to claims 1-30 have been considered but are moot in view of the new ground(s) of rejection.

### ***Conclusion***

20. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date

Art Unit: 2153

of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to PHILIP J. CHEA whose telephone number is (571)272-3951. The examiner can normally be reached on M-F 6:30-4:00 (1st Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenn Burgess can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Philip J Chea  
Examiner  
Art Unit 2153

PJC 3/4/08

/Glenton Burgess/

Supervisory Patent Examiner, Art Unit 2153